



DoaSô

**Ajude hoje e transforme o
amanhã**

Integrantes

Gabriel Madureira Matos

Guilherme de Oliveira Souza

Igor Maia Moreira Fernandes

Luis Henrique Sampaio Vieira

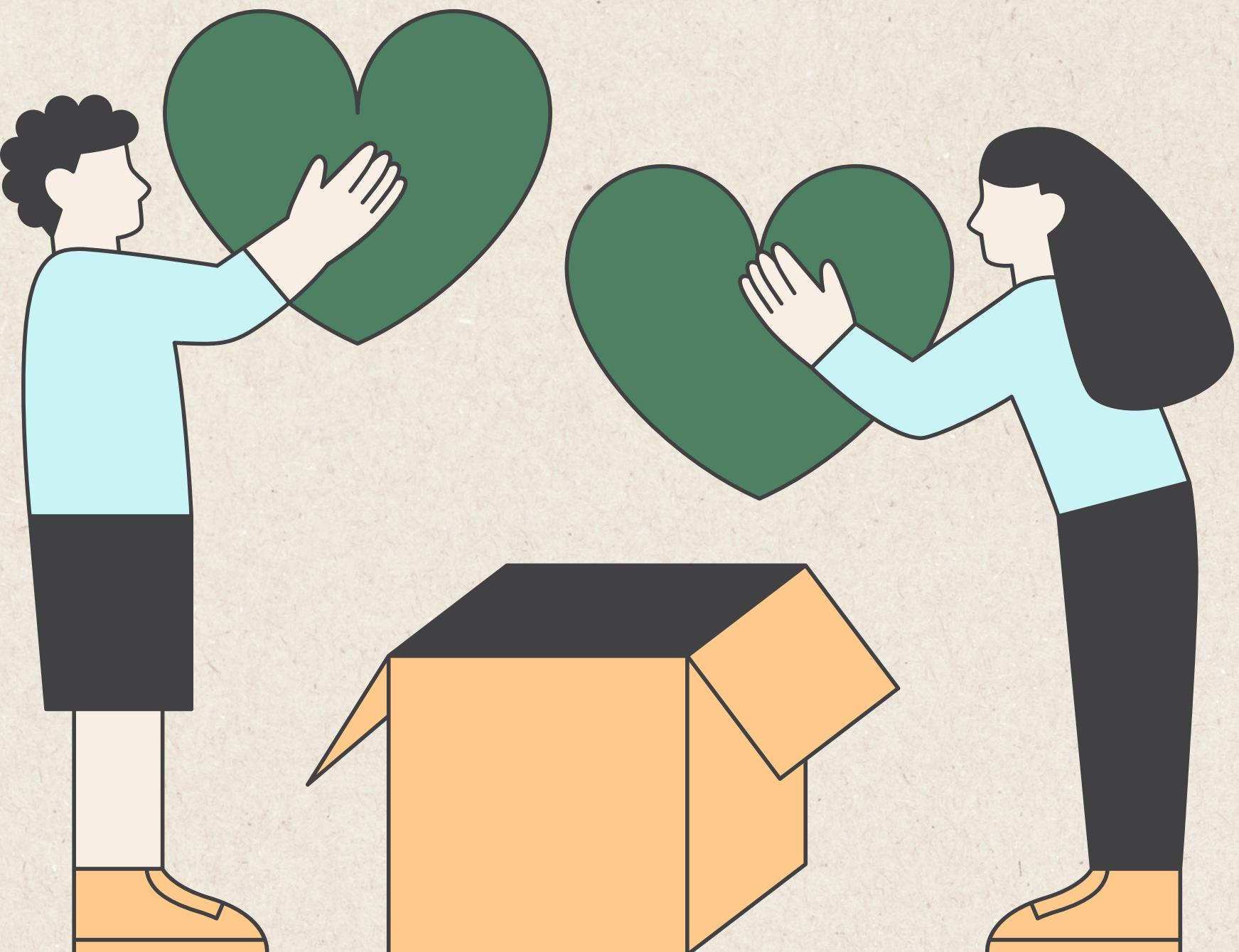
Thiago Moreira de Queiroz

Victor Schneider do Vale

Vitor Prates Souza Lima

Wendell Leonardo Machado

Miranda



DoaSô

- Solução Tecnológica e Sustentável
- Eficiência e Transparência no Processo
- Impacto Social e Ambiental

Credenciais de Acesso

```
const mysql = require('mysql2')

const db = mysql.createConnection({
  host: "localhost",
  user: "root",
  password: "doaso123",
  database: "doaso",
  port: 3306
})

db.connect((err) => {
  if (err) {
    console.log("Falha de conexão! " + err.stack);
    return;
  }

  console.log("Conexão estabelecida: " + db.threadId);
});

module.exports = db;
```

Organização dos Módulos

```
✓ SRC
  > back
  > db
  > front
```

```
✓ front
  > Cadastro
  > CriarMetas
  > DetalhamentoDeMetas
  > Home
  > images
  > Login
  > Navegação
  > Perfil
  ✓ script
    JS Metas.js
    JS Proposta.js
    JS Usuario.js
    # global.css
    JS global.js
```

```
✓ back
  > node_modules
  JS connection_mysql.js
  {} exemploRequisicoes.json
  JS index.js
  {} package-lock.json
  {} package.json
  i README.md
  ✓ db
  i README.md
  SQLAtualizado.sql
```

Rotas de requisição HTTP

Requisição	Query no Banco de Dados (MySQL)
GET	SELECT * FROM Nome_da_Tabela
POST	INSERT INTO Nome_da_Tabela (id) VALUES (1)
PUT	UPDATE Nome_da_Tabela SET descricao = "exemplo id 1"
DELETE	DELETE FROM Nome_da_Tabela WHERE id= 1

Código requisição

```
//GET Metas de doação por id
app.get("/api/meta/:id", (req, res) => {
  const idMeta = req.params.id
  db.query("SELECT * FROM Meta_de_doacao WHERE id_meta = ?", idMeta, (err, result) => {
    if (err) {
      console.log(err)
    }
    res.send(result)
  });
});
```

```
//PUT metas de doação
app.put('/api/meta/:id', (req, res) => {
  var id = req.params.id;
  const { valorObjetivo, valorArrecadado, descricao, titulo } = req.body;

  db.query("UPDATE Meta_de_doacao SET valor_objetivo_meta = ?, valor_recebido_meta = ?, desc_meta = ?,
  titulo_meta = ? WHERE id_meta = ?",
  [valorObjetivo, valorArrecadado, descricao, titulo, id], (err, result) => {
    if (err) { ...
    } else { ...
    }
  });
});
```

```
//POST Metas de Doacao
app.post('/api/meta', (req, res) => {
  const { valorObjetivo, valorArrecadado, descricao, titulo, idCentroCriador} = req.body;

  db.query("INSERT INTO Meta_de_doacao(valor_objetivo_meta, valor_recebido_meta, desc_meta,
  titulo_meta, id_centro_criador) VALUES (?, ?, ?, ?, ?)", [valorObjetivo, valorArrecadado, descricao,
  titulo, idCentroCriador], (err, result) => {
    if (err) { ...
    }
  });
});
```

```
//DELETE metas de doação
app.delete('/api/meta/:id', (req, res) => {
  id = req.params.id;

  db.query("DELETE FROM Meta_de_doacao WHERE id_meta = ?", [id], (err, result) => {
    if (err) {
      console.log(err);
      return res.status(500).json({ error: 'Erro ao deletar meta' });
    }
  });
});
```

Classes do sistema

Classe	Métodos
Usuario.js	Contem os métodos CRUD dos doadores e dos centros de doação, incluindo leituras completas do banco e leituras filtradas por ID
Metas.js	Contem os métodos CRUD para as metas de doação, incluindo leituras completas do banco e leituras filtradas por ID
Proposta.js	Contem os métodos CREATE, READ e DELETE para as metas de doação, incluindo leituras completas do banco e leituras filtradas por ID. Esse método não possui método UPDATE pois não pode ser atualizado
ImagenDoacao.js	Contem os métodos CREATE e READ das imagens anexadas nas propostas de doação
Mensagem.js	Contem os métodos CRUD para as mensagens do chat entre os doadores e centros de doação, tendo as leituras filtradas pelo id da proposta do chat.

Códigos Classes

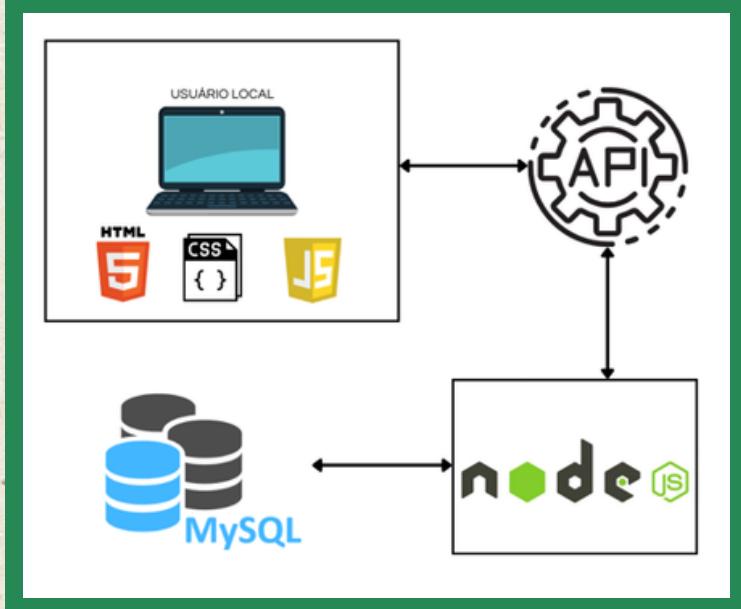
Classe Metas.js

```
JS Metas.js  X
src > front > script > JS Metas.js > [o] default
1  const url = "http://localhost:3307/api/meta";
2
3  const MetaHelper = {
4    getMeta: async function(){ ... },
5    getMetaById: async function(id){ ... },
6    postMeta: async function(body){ ... },
7    putMeta: async function(id, body){ ... },
8    deleteMeta: async function(id){ ... }
9
10 }
11
12 export default MetaHelper;
```

Método getMeta()

```
JS Metas.js  X
src > front > script > JS Metas.js > [o] default
1  const url = "http://localhost:3307/api/meta";
2
3  const MetaHelper = {
4    getMeta: async function(){
5      return fetch(url, {
6        method: "GET",
7        headers: {
8          "Content-Type": "application/json"
9        }
10       })
11     .then(response => {
12       if (!response.ok) {
13         throw new Error(`Erro na requisição: ${response.status}`);
14       }
15       return response.json();
16     })
17     .then(responseData => {
18       console.log("Resposta do servidor:", responseData);
19       return responseData;
20     })
21     .catch(error => {
22       console.error("Erro ao fazer a requisição GET:", error);
23       throw error;
24     });
25   },
```

Arquitetura



```
import MetaHelper from "../../script/Metas.js";

var btn = document.getElementById('btnCriarMeta');

function postarMeta() {
  var valorObjetivo = document.getElementById('valorObjetivo').value;
  var titulo = document.getElementById('tituloMeta').value;
  var descricao = document.getElementById('descricaoMeta').value;

  const body = {
    // ...
  }

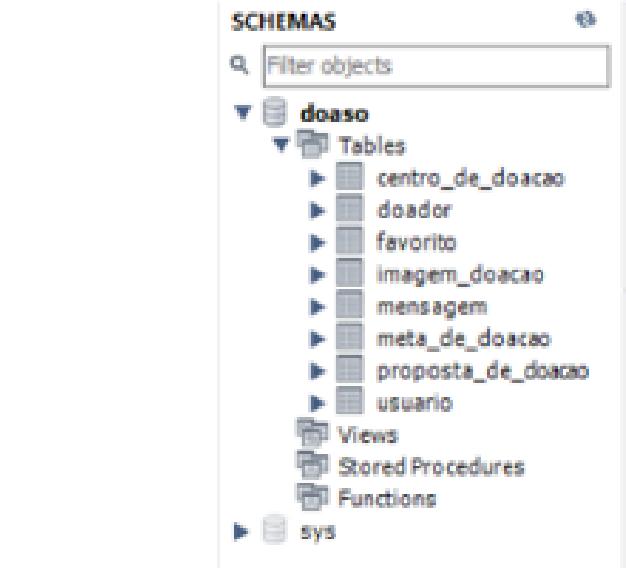
  MetaHelper.postMeta(body);
}

btn.addEventListener('click', postarMeta);
```

```
const fetch = require('node-fetch');
const express = require('express');
const app = express();
const port = 3001;

const MetasHelper = {
  getMeta: async function() {
    return fetch('http://localhost:3002/api/meta')
      .then(response => {
        if (!response.ok) {
          throw new Error(`Erro na requisição: ${response.status}`);
        }
        return response.json();
      })
      .then(respostaData => {
        console.log("Resposta do servidor:", respostaData);
        return respostaData;
      })
      .catch(error => {
        console.error("Erro ao fazer a requisição API", error);
        throw error;
      });
  }
};

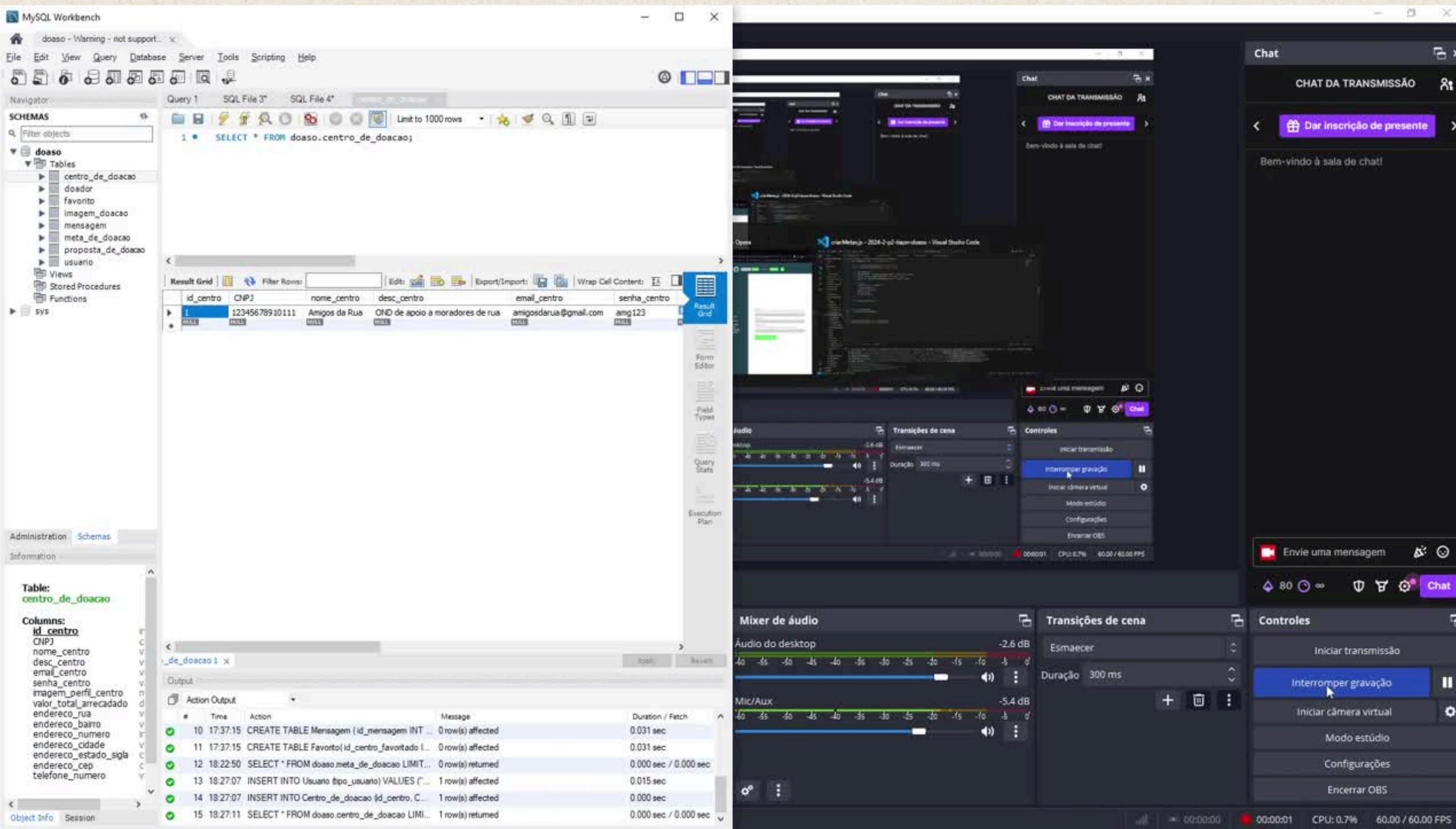
app.get('/api/meta/:id', (req, res) => {
  const idMeta = req.params.id;
  db.query("SELECT * FROM Meta_de_doacao WHERE id_meta = ?", idMeta, (err, result) => {
    if (err) {
      console.log(err);
    }
    res.send(result);
  });
});
```

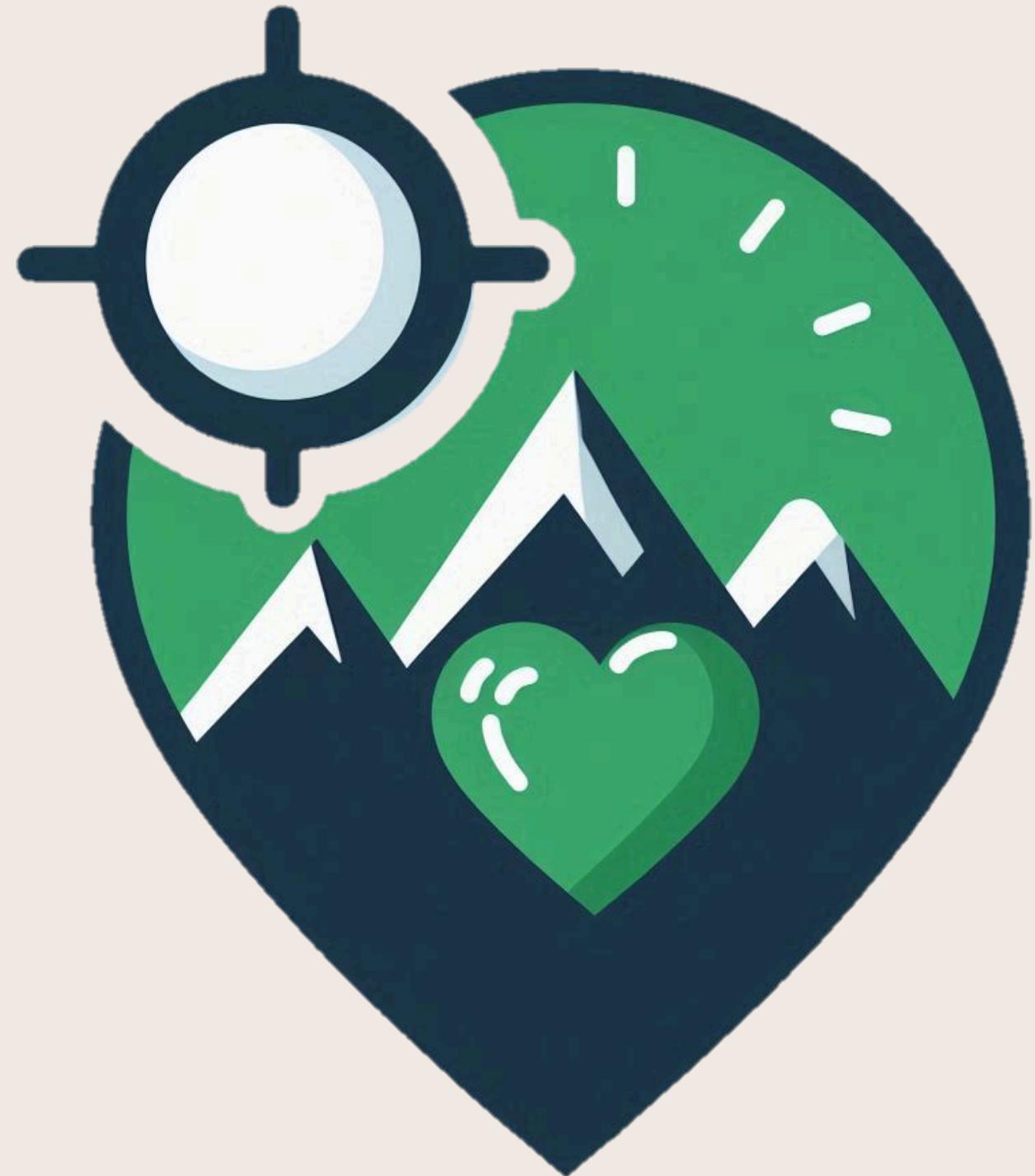


```
//GET Metas de doação por id
app.get("/api/meta/:id", (req, res) => {
  const idMeta = req.params.id;
  db.query("SELECT * FROM Meta_de_doacao WHERE id_meta = ?", idMeta, (err, result) => {
    if (err) {
      console.log(err);
    }
    res.send(result);
  });
});
```

Exemplo Prático

Cadastro e remoção de uma meta no banco de dados:





OBRIGADO!